

What is claimed is:

5 1. A method for managing compression of pages of memory in a system comprising physical memory, wherein the physical memory comprises system memory, the method comprising:

receiving a system memory access;

locating a page translation entry for the system memory access in a page translation table;

10 determining if a page in the physical memory and referenced by the page translation entry is compressed or uncompressed;

if said determining indicates the page is compressed:

decompressing the compressed page to produce a decompressed page;

writing the decompressed page to the physical memory; and

5 providing a first physical memory address of the decompressed page in the physical memory to fulfill the system memory access.

2. The method of claim 1, further comprising:

if said determining indicates the page is uncompressed, providing a second physical memory address as indicated by the page translation entry to fulfill the system memory access.

3. The method of claim 1, further comprising:

if said determining indicates the page is uncompressed:

determining if the uncompressed page is to be compressed;

25 if said determining indicates the page is to be compressed:

compressing the uncompressed page to produce a compressed page;

and

writing the compressed page to the physical memory.

Sub
A3

4. The method of claim 3, wherein said compressing the uncompressed page comprises:

providing the uncompressed page to a compression engine; and
the compression engine compressing the page to produce the compressed page.

5

5. The method of claim 4, wherein said providing the page to the compression engine comprises:

a Direct Memory Access (DMA) channel reading the uncompressed page from the physical memory; and

10

the DMA channel writing the uncompressed page to the compression engine.

6. The method of claim 4, wherein said writing the compressed page to the physical memory comprises:

a Direct Memory Access (DMA) channel reading the compressed page from the compression engine; and

15

the DMA channel copying the compressed page into one or more linked compressed blocks in the physical memory.

7. The method of claim 6, further comprising locating the one or more linked compressed blocks for storing the compressed page in a list of available compressed blocks for storing compressed pages.

20

8. The method of claim 4, wherein said providing the page to the compression engine comprises:

25

a plurality of Direct Memory Access (DMA) channels reading the uncompressed page from the physical memory; and

the plurality of DMA channels writing the uncompressed page to the compression engine.

Sub
A3

9. The method of claim 4, wherein said writing the compressed page to the physical memory comprises:

a plurality of Direct Memory Access (DMA) channels reading the compressed page from the compression engine; and

5 the plurality of DMA channels copying the compressed page into one or more linked compressed blocks in the physical memory.

10. The method of claim 3, wherein said compressing the uncompressed page comprises:

10 providing a different portion of the uncompressed page to each of a plurality of compression engines; and

the plurality of compression engines compressing the provided uncompressed portions of the page to produce compressed portions of the page.

15 11. The method of claim 10, wherein each of the plurality of compression engines implements a data compression algorithm, wherein the data compression algorithm is substantially the same for each of the plurality of compression engines.

20 12. The method of claim 10, wherein the plurality of compression engines compresses the uncompressed portions of the compressed page in parallel.

13. The method of claim 10, further comprising combining the compressed portions of the page to produce the compressed page.

25 14. The method of claim 3, wherein said compressing the uncompressed page comprises:

providing the uncompressed page to a plurality of compression engines, wherein each of the plurality of compression engines implements a different compression algorithm;

Sub
A3

the plurality of compression engines each compressing the uncompressed page using the compression algorithm implemented by the particular compression engine to produce a plurality of compressed pages compressed by different compression algorithms;

5 selecting the compressed page from the plurality of compressed pages, wherein the selected compressed page has the highest compression ratio of the plurality of compressed pages.

10 15. The method of claim 14, further comprising marking the page translation entry associated with the selected compressed page to indicate the particular compression algorithm used in said compressing the page.

16. The method of claim 14, wherein the plurality of compression engines compresses the page in parallel.

15 17. The method of claim 1, wherein one or more recently used page translation entries from the page translation table are cached in a page translation cache, and wherein said locating a page translation entry comprises:

searching for the page translation entry associated with the system memory address in the page translation cache;

20 wherein, if said searching locates the page translation entry in the page translation cache, the page translation entry from the page translation cache is used in said determining if the page is compressed or uncompressed.

25 18. The method of claim 17, wherein, if the page translation entry is not located in the page translation cache, the method further comprises searching for the page translation entry in the page translation table;

wherein, if said searching locates the page translation entry in the page translation table, the page translation entry from the page translation table is used in said determining if the page is compressed or uncompressed.

Sub
A3

19. The method of claim 18, further comprising caching the page translation entry located in the page translation table to the page translation cache as a recently used page translation entry.

5 20. The method of claim 17, wherein the page translation cache comprises a plurality of page translation entries, and wherein the page translation cache is fully associative.

10 21. The method of claim 20, wherein said searching for the page translation entry associated with the system memory address in the page translation cache comprises comparing the system memory address with all page translation entries in the page translation cache in parallel.

22. The method of claim 1, wherein said decompressing the page comprises:
5 providing the page to a decompression engine; and
the decompression engine decompressing the page to produce the decompressed page.

23. The method of claim 22, further comprising, if said determining indicates
20 the page is compressed:

prior to said providing the page to a decompression engine:

examining the page translation entry to determine a compression algorithm
used to compress the page; and

25 selecting the decompression engine from a plurality of decompression engines, wherein the decompression engine is configured to decompress data compressed using the determined compression algorithm.

30 24. The method of claim 22, wherein the decompression engine is a parallel decompression engine, wherein, in said decompressing the compressed page, the parallel decompression engine decompresses portions of the page in parallel.

Sub
A3 >

25. The method of claim 22, wherein the page comprises one or more compressed blocks, and wherein said providing the page to the decompression engine comprises:

5 a Direct Memory Access (DMA) channel reading the one or more compressed blocks; and

the DMA channel copying the one or more compressed blocks to the decompression engine.

10 26. The method of claim 25, wherein the DMA channel reading the one or more compressed blocks comprises:

loading a physical memory address of a first compressed block into the DMA channel;

reading the first compressed block into the DMA channel; and

15 reading one or more subsequent compressed blocks linked to the first compressed block into the DMA channel.

20 27. The method of claim 25, wherein said writing the decompressed page to the physical memory is performed by the DMA channel, and wherein said writing the decompressed page to the physical memory comprises the DMA channel reading the uncompressed page from the decompression engine.

25 28. The method of claim 22, wherein the page comprises one or more compressed blocks, and wherein said providing the page to the decompression engine comprises:

a plurality of Direct Memory Access (DMA) channels reading the one or more compressed blocks; and

the plurality of DMA channels copying the one or more compressed blocks to the decompression engine.

Sub
A3

29. The method of claim 1, further comprising:
providing a different portion of the compressed page to each of a plurality of
decompression engines; and

each of the plurality of decompression engines decompressing the portion of the
compressed page provided to the particular decompression engine.

30. The method of claim 29, wherein each of the plurality of decompression
engines implements a data decompression algorithm, wherein the data decompression
algorithm is substantially the same for each of the plurality of decompression engines.

31. The method of claim 29, wherein the plurality of decompression engines
decompresses the portions of the compressed page in parallel.

32. The method of claim 29, further comprising combining the decompressed
portions of the page to produce the decompressed page.

33. The method of claim 1, wherein said writing the decompressed page to the
physical memory comprises:

locating a currently unused page in the physical memory in a list of currently unused
pages for receiving uncompressed pages; and

writing the decompressed page as an uncompressed page to the located currently
unused page.

34. The method of claim 1, wherein the compression of pages of the memory in
the system is operable to increase the effective size of the system memory by keeping least
recently used data as compressed data in the physical memory and most recently and
frequently used data as uncompressed data in the physical memory.

Sub
A3

35. The method of claim 34, wherein the system further comprises an operating system, wherein the operating system is aware of the increased effective size of the system memory.

5 36. The method of claim 34, wherein the system further comprises an operating system, wherein the operating system is not aware of the increased effective size of the system memory.

10 37. The method of claim 34, wherein the system further comprises an operating system, wherein the operating system is aware of the increased effective size of a first portion of the system memory, and wherein the operating system is not aware of the increased effective size of a second portion of the system memory.

15 38. The method of claim 1, wherein said decompressing the compressed page to produce the decompressed page comprises:

examining the page translation entry to determine if the page translation entry indicates the page is highly compressed, wherein, during compression of the page to generate the highly compressed page, the physical memory occupied by the page is freed for use by one or more processes executing within the system; and

20 if said examining determines the page is highly compressed:

allocating a portion of the physical memory for the page; and

writing data stored prior to said compressing the page to the page.

25 39. The method of claim 38, wherein the page translation entry includes a highly compressed attribute field, wherein the highly compressed attribute field indicates if the page is highly compressed, wherein said examining the page translation entry to determine if the page is highly compressed comprises examining the highly compressed attribute field.

30 40. The method of claim 39, wherein the highly compressed attribute field is a one-bit field.

Sub
A3

41. A method for managing compression of pages of memory in a system comprising an operating system and physical memory, wherein the physical memory comprises system memory, the method comprising:

locating a page translation entry in a page translation table, wherein the page translation entry references an uncompressed page in the physical memory;

determining if the uncompressed page is to be compressed;

if said determining indicates the page is to be compressed:

compressing the uncompressed page to produce a compressed page; and

writing the compressed page to the physical memory.

wherein the compression of pages of the memory in the system is operable to increase the effective size of the system memory by keeping least recently used data as compressed data in the physical memory and most recently and frequently used data as uncompressed data in the physical memory; and

wherein the operating system is not aware of the increased effective size of the system memory.

42. The method of claim 41, wherein said compressing the uncompressed page comprises:

providing the uncompressed page to a compression engine; and

the compression engine compressing the page to produce the compressed page.

43. The method of claim 42, wherein said providing the page to the compression engine comprises:

one or more Direct Memory Access (DMA) channels reading the uncompressed page from the physical memory; and

the one or more DMA channels writing the uncompressed page to the compression engine.

Sub
A2

44. The method of claim 42, wherein said writing the compressed page to the physical memory comprises:

one or more Direct Memory Access (DMA) channels reading the compressed page from the compression engine; and

the one or more DMA channels copying the compressed page into one or more linked compressed blocks in the physical memory.

45. The method of claim 44, further comprising locating the one or more linked compressed blocks for storing the compressed page in a list of available compressed blocks for storing compressed pages.

46. The method of claim 41, wherein said compressing the uncompressed page comprises:

providing a different portion of the uncompressed page to each of a plurality of compression engines; and

the plurality of compression engines each compressing the portion of the page which the particular compression engine was provided; and

combining the compressed portions of the page to produce the compressed page.

47. The method of claim 46, wherein the plurality of compression engines compress the portions of the compressed page in parallel.

48. The method of claim 41, wherein said compressing the uncompressed page comprises:

providing the uncompressed page to a plurality of compression engines, wherein each of the plurality of compression engines implements a different compression algorithm;

the plurality of compression engines each compressing the uncompressed page using the compression algorithm implemented by the particular compression engine to produce a plurality of compressed pages each compressed by a different compression algorithm;

Sub
A3

selecting the compressed page from the plurality of compressed pages, wherein the selected compressed page has the highest compression ratio of the plurality of compressed pages.

5

49. The method of claim 48, further comprising marking the page translation entry associated with the compressed page to indicate the particular compression algorithm used in said compressing the page.

10

50. A method for compressing memory in a system comprising a plurality of compression engines and a physical memory, wherein the physical memory comprises system memory, the method comprising:

locating a page translation entry in a page translation table, wherein the page translation entry references an uncompressed page in the physical memory;

providing the referenced uncompressed page to the plurality of compression engines, wherein each of the plurality of compression engines implements a different compression algorithm;

the plurality of compression engines each compressing the uncompressed page using the compression algorithm implemented by the particular compression engine to produce a plurality of compressed pages each compressed by a different compression algorithm;

selecting the compressed page with the highest compression ratio of the plurality of compressed pages; and

writing the selected compressed page to the physical memory.

25

51. The method of claim 50, further comprising marking the page translation entry associated with the selected compressed page to indicate the particular compression algorithm used in said compressing the page.

30

52. The method of claim 50, further comprising:

Sub
R3

determining that the compressed page needs to be decompressed;
examining the page translation entry to determine the particular compression
algorithm used to compress the page;
selecting a decompression engine from a plurality of decompression engines,
5 wherein the selected decompression engine implements a decompression algorithm for
decompressing data compressed using the particular compression algorithm;
providing the page to the selected decompression engine; and
the selected decompression engine decompressing the page using the decompression
algorithm to produce the decompressed page.

10

53. The method of claim 50, wherein the compression of pages of the memory
in the system is operable to increase the effective size of the system memory by keeping
least recently used data as compressed data in the physical memory and most recently and
frequently used data as uncompressed data in the physical memory.

5
10
15
20

54. The method of claim 53, wherein the system further comprises an operating
system, wherein the operating system is not aware of the increased effective size of the
system memory.

55. A method comprising:
determining a compression ratio for system memory in a system comprising
physical memory, wherein the physical memory comprises the system memory;
determining if the compression ratio is below a compression ratio threshold;
25 if the compression ratio is below the compression ratio threshold:
locating a page translation entry in a page translation table referencing an
uncompressed page in the system memory to be highly compressed;
setting a highly compressed attribute in the page translation entry to indicate
that the page is highly compressed; and

25

Sub
A3

freeing a first portion of the physical memory allocated to the page in the system memory;

wherein highly compressing the page of the memory in the system is operable to increase the compression ratio for the system memory.

5

56. The method of claim 55, wherein the highly compressed attribute is a one-bit field in the page translation entry.

10

57. The method of claim 55, further comprising:
receiving a system memory access for the page, wherein the system memory access requires that the page be uncompressed;

locating the page translation entry in the page translation table referencing the page;
determining that the page is highly compressed; and
allocating a second portion of the physical memory for the page.

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

58. The method of claim 57, further comprising filling the page with zeros after said allocating.

59. The method of claim 57, further comprising, after said allocating:
reading data from non-volatile storage; and
writing the data to the page.

60. The method of claim 55, further comprising, prior to said freeing the physical memory allocated to the page:

determining if the page is dirty; and

if the page is dirty, writing data from the page to non-volatile storage to make the page clean.

30

61. A system comprising:

Sub
A3

one or more processors;
a physical memory comprising a system memory;
a system memory controller; and
a compressed memory management unit (CMMU), configured to:

5 receive from a first processor of the one or more processors a system
memory access comprising a system memory address;
translate the system memory address into a first physical memory address;
cause the decompression of the compressed data at the first physical memory
address in the physical memory;
10 write the decompressed data to a second physical memory address; and
pass the second physical memory address to the system memory controller;
wherein the system memory controller is configured to fulfill the system memory
access from the decompressed data at the second physical memory address; and
wherein the CMMU is operable to increase the effective size of the system memory
5 by keeping least recently used data as compressed data in the physical memory and most
recently and frequently used data as uncompressed data in the physical memory.

62. The system of claim 61, wherein the system further comprises program
instructions executable within the system to implement an operating system, wherein the
operating system is aware of the increased effective size of the system memory.
20

63. The system of claim 61, wherein the system further comprises program
instructions executable within the system to implement an operating system, wherein the
operating system is not aware of the increased effective size of the system memory.
25

64. The system of claim 61, wherein the system further comprises program
instructions executable within the system to implement an operating system, wherein the
operating system is aware of the increased effective size of a first portion of the system
memory, and wherein the operating system is not aware of the increased effective size of a
30 second portion of the system memory.

Sub
A3

65. The system of claim 61, wherein the CMMU comprises:
a page translation cache configured to cache page translation entries; and
one or more scatter/gather Direct Memory Access (DMA) channels configured for
transferring data from the CMMU to one or more destinations and for receiving data on the
CMMU from one or more sources.

66. The system of claim 17, wherein the CMMU further comprises a
compression/decompression engine configured to compress uncompressed data and to
decompress compressed data under control of the CMMU.

67. The system of claim 61, further comprising:
a page translation table comprising one or more page translation entries;
wherein, in said translating the system memory address into a first physical memory
address, the CMMU is further configured to:

locate a page translation entry for the system memory address in the page
translation table; and

determine the first physical memory address from the page translation entry
for the system memory address.

68. The system of claim 67, wherein the CMMU further comprises:
a page translation cache comprising one or more cached page translation entries
from the page translation table;

wherein, in said locating a page translation entry, the CMMU is further configured
to search for the page translation entry associated with the system memory address in the
page translation cache;

wherein, if said searching the page translation cache locates the page translation
entry in the page translation cache, the page translation entry from the page translation
cache is used in said determining the first physical memory address.

Sub
A3 }
69. The system of claim 68, wherein, if said searching the page translation cache does not locate the page translation entry in the page translation cache, the CMMU is further configured to:

search for the page translation entry in the page translation table;

5 wherein, if said searching the page translation table locates the page translation entry in the page translation table, the page translation entry from the page translation table is used in said determining the first physical memory address.

10 70. The system of claim 69, wherein the CMMU is further configured to cache the page translation entry located in the page translation table to the page translation cache as a recently used page translation entry.

15 71. The system of claim 68, wherein the page translation cache is fully associative.

20 72. The system of claim 71, wherein, in said searching for the page translation entry associated with the system memory address in the page translation cache, the CMMU is further configured to compare the system memory address with all page translation entries in the page translation cache in parallel.

73. The system of claim 61, wherein the system further comprises:

a compression/decompression engine;

25 wherein, in said causing the decompression of the compressed data, the CMMU is further configured to write the compressed data to the compression/decompression engine; and

wherein the compression/decompression engine is configured to decompress the compressed data to produce the decompressed data.

74. The system of claim 73, wherein the CMMU further comprises:

30 one or more Direct Memory Access (DMA) channels;

Sub
R3

wherein the compressed data comprises one or more compressed blocks, and wherein, in said writing the compressed data to the compression/decompression engine, the one or more DMA channels are configured to:

read the one or more compressed blocks; and

copy the one or more compressed blocks to the compression/decompression engine.

75. The system of claim 74, wherein the CMMU is further configured to:

load a physical memory address of a first compressed block into the one or more DMA channels; and

wherein in reading the one or more compressed blocks, the one or more DMA channels are further configured to read the first compressed block and one or more subsequent compressed blocks linked to the first compressed block.

76. The system of claim 74, wherein said writing the decompressed data to the second physical memory address is performed by the one or more DMA channels, and wherein, in said writing the decompressed data to the second physical memory address, the one or more DMA channels are further configured to read the decompressed data from the compression/decompression engine.

77. The system of claim 61, wherein the system further comprises a plurality of compression/decompression engines, wherein at least two of the plurality of compression/decompression engines implement different compression/decompression algorithms, and wherein the CMMU is further configured to:

prior to said writing the compressed data to the compression/decompression engine:

determine a particular compression algorithm used to compress the compressed data; and

select the compression/decompression engine from the plurality of compression/decompression engines, wherein the selected compression/decompression

Sub
P3

engine is configured to decompress data compressed using the particular compression algorithm.

5

78. The system of claim 61, wherein the system further comprises:
a compression/decompression engine configured to decompress compressed data under control of the CMMU.

10

79. The system of claim 78, wherein the compression/decompression engine is a parallel compression/decompression engine configured to perform parallel data decompression under control of the CMMU.

15
20

80. The system of claim 78, wherein the compression/decompression engine is comprised in the CMMU.

81. The system of claim 61, wherein the CMMU is comprised in one of the one or more processors.

82. The system of claim 81, wherein at least one of the one or more processors further comprises a compression/decompression engine configured to decompress compressed data under control of the CMMU.

25

83. The system of claim 81, wherein the system memory controller comprises a compression/decompression engine configured to decompress compressed data under control of the CMMU.

30

84. The system of claim 81, wherein the physical memory comprises one or more memory modules, and wherein at least one of the one or more memory modules comprises a compression/decompression engine configured to decompress compressed data under control of the CMMU.

Sub
A3
85. The system of claim 61, wherein the CMMU is comprised in the system memory controller.

86. The system of claim 85, wherein the system memory controller further comprises a compression/decompression engine configured to decompress compressed data under control of the CMMU.

87. The system of claim 85, wherein the physical memory comprises one or more memory modules, and wherein at least one of the one or more memory modules comprises a compression/decompression engine configured to decompress compressed data under control of the CMMU.

88. The system of claim 61, wherein the physical memory comprises one or more memory modules, wherein the CMMU is coupled to the system memory controller and the one or more memory modules.

89. The system of claim 88, wherein at least one of the one or more memory modules comprises a compression/decompression engine configured to decompress compressed data under control of the CMMU.

90. The system of claim 61, wherein the system further comprises:
a plurality of compression/decompression engines;
wherein, in said causing the decompression of the compressed data, the CMMU is further configured to write a different portion of the compressed data to each of the plurality of compression/decompression engines;

wherein the plurality of compression/decompression engines are configured to decompress the portions of the compressed data to produce a plurality of decompressed data portions; and

wherein the CMMU is further configured to combine the plurality of decompressed data portions to produce the decompressed data.

Sub
A3

91. The system of claim 90, wherein each of the plurality of compression/decompression engines implements a data decompression algorithm, wherein the data decompression algorithm is substantially the same for each of the plurality of compression/decompression engines.

92. The system of claim 90, wherein the plurality of compression/decompression engines decompress the portions of the compressed data in parallel.

93. The system of claim 61, wherein, in said writing the decompressed data to physical memory, the CMMU is further configured to:

locate a currently unused page in the physical memory in a list of currently unused pages for receiving uncompressed data; and

write the decompressed data as an uncompressed page to the located currently unused page.

94. The system of claim 61, wherein the CMMU is further configured to manage the system memory on a page granularity.

95. The system of claim 61, wherein page size is programmable.

96. The system of claim 61, wherein a maximum compression ratio applied by the CMMU is programmable.

97. The system of claim 61, further comprising a kernel driver configured to: monitor an actual compression ratio achieved by the CMMU; and ensure that a minimum compression ratio is maintained by the CMMU.

Sub
R3

98. A system comprising:

one or more processors;

a system memory controller;

a physical memory comprising a system memory; and

a compressed memory management unit (CMMU), configured to:

translate a system memory address into a first physical memory address;

cause the compression of the uncompressed data at the first physical memory address to produce compressed data; and

write the compressed data to a second physical memory address;

wherein the system is operable to increase the effective size of system memory by keeping least recently used pages compressed in the physical memory and most recently and frequently used pages uncompressed in the physical memory.

99. The system of claim 98, wherein the system further comprises program instructions executable within the system to implement an operating system, wherein the operating system is aware of the increased effective size of the system memory.

100. The system of claim 98, wherein the system further comprises program instructions executable within the system to implement an operating system, wherein the operating system is not aware of the increased effective size of the system memory.

101. The system of claim 98, wherein the system further comprises program instructions executable within the system to implement an operating system, wherein the operating system is aware of the increased effective size of a first portion of the system memory, and wherein the operating system is not aware of the increased effective size of a second portion of the system memory.

102. The system of claim 98, further comprising:
wherein the CMMU is further configured to receive from a first processor of the one or more processors the system memory access comprising a system memory address.

Sub
B3

103. The system of claim 98, wherein the CMMU comprises:

a page translation cache configured to cache page translation entries; and

one or more scatter/gather Direct Memory Access (DMA) channels configured for

transferring data from the CMMU to one or more destinations and for receiving data on the
CMMU from one or more sources.

104. The system of claim 103, wherein the CMMU further comprises a

compression/decompression engine configured to compress uncompressed data and to
decompress compressed data under control of the CMMU.

105. The system of claim 103, wherein the page translation cache is fully

associative.

106. The system of claim 98, wherein the system further comprises:

a compression/decompression engine configured to compress uncompressed data

under control of the CMMU.

107. The system of claim 106, wherein the compression/decompression engine is

a parallel compression/decompression engine configured to perform parallel data
compression under control of the CMMU.

108. The system of claim 106, wherein the compression/decompression engine is

comprised in the CMMU.

109. The system of claim 98, wherein the CMMU is comprised in one of the one

or more processors.

110. The system of claim 109, wherein at least one of the one or more processors

comprises a compression/decompression engine configured to compress uncompressed data
under control of the CMMU.

Sub
A3

111. The system of claim 109, wherein the system memory controller comprises a compression/decompression engine configured to compress uncompressed data under control of the CMMU.

5

112. The system of claim 109, wherein the physical memory comprises one or more memory modules, and wherein the at least one of the one or more memory modules comprises a compression/decompression engine configured to compress uncompressed data under control of the CMMU.

10

113. The system of claim 98, wherein the CMMU is comprised in the system memory controller.

114. The system of claim 113, wherein the system memory controller further comprises a compression/decompression engine configured to compress uncompressed data under control of the CMMU.

115. The system of claim 113, wherein the physical memory comprises one or more memory modules, and wherein at least one of the one or more memory modules comprises a compression/decompression engine configured to compress uncompressed data under control of the CMMU.

116. The system of claim 98, wherein the physical memory comprises one or more memory modules, wherein the CMMU is coupled to the system memory controller and the one or more memory modules.

117. The system of claim 116, wherein at least one of the one or more memory modules comprises a compression/decompression engine configured to compress uncompressed data under control of the CMMU.

30

Sub
R3

118. The system of claim 98, wherein the system further comprises:

a plurality of compression/decompression engines;

wherein, in said causing the compression of the uncompressed data, the CMMU is further configured to write a different portion of the uncompressed data to each of the plurality of compression/decompression engines;

wherein the plurality of compression/decompression engines are configured to compress the portions of the uncompressed data to produce a plurality of compressed data portions; and

wherein the CMMU is further configured to combine the plurality of compressed data portions to produce the compressed data.

119. The system of claim 118, wherein each of the plurality of compression/decompression engines implements a data compression algorithm, wherein the data compression algorithm is substantially the same for each of the plurality of compression/decompression engines.

120. The system of claim 118, wherein the plurality of compression/decompression engines is configured to compress the portions of the uncompressed data in parallel.

121. The system of claim 98, wherein the system further comprises:

a plurality of compression/decompression engines, wherein each of the plurality of compression/decompression engines implements a different compression algorithm;

wherein, in said causing the compression of the uncompressed data, the CMMU is further configured to provide the uncompressed data to each of the plurality of compression/decompression engines;

wherein the plurality of compression engines are configured to each compress the uncompressed data using the compression algorithm implemented by the particular compression engine to produce a plurality of compressed data each compressed by a different compression algorithm; and

but
A3

wherein, in said causing the compression of the uncompressed data, the CMMU is further configured to select the compressed data from among the plurality of compressed data.

5 122. The system of claim 121, wherein the CMMU selects the compressed data with the highest compression ratio from among the plurality of compressed data.

10 123. The system of claim 121, wherein the system further comprises a page translation table comprising one or more page translation entries, wherein one of the one or more page translation entries references a page of physical memory at the second physical memory address, wherein the CMMU is further configured to mark the page translation entry to indicate the particular compression algorithm used in said compressing the uncompressed data.

15 124. The system of claim 121, wherein the plurality of compression/decompression engines is configured to compress the uncompressed data in parallel.

20 125. The system of claim 98, further comprising:
a compression/decompression engine;
wherein, in said causing the compression of the uncompressed data, the CMMU is further configured to write the uncompressed data to the compression/decompression engine; and

25 wherein the compression engine is configured to compress the uncompressed data to produce the compressed data.

30 126. The system of claim 125, wherein the CMMU further comprises:
one or more Direct Memory Access (DMA) channels;
wherein, in said writing the uncompressed data to the compression/decompression engine, the one or more DMA channels are configured to:

Sub
A3

read the uncompressed data from physical memory; and
write the uncompressed data to the compression/decompression engine.

127. The system of claim 125, wherein, in said writing the compressed data to the
5 second physical memory address, the one or more DMA channels are further configured to:
read the compressed page from the compression/decompression engine; and
copy the compressed data into one or more linked compressed blocks in physical
memory.

128. The system of claim 125, wherein the CMMU is further configured to:
10 locate the one or more compressed blocks for storing the compressed data in a list of
available compressed blocks for storing compressed data; and
link the one or more compressed blocks to generate the one or more linked
compressed blocks.

129. A system comprising:
one or more processors;
a system memory controller;
a physical memory comprising a system memory; and
20 a compressed memory management unit (CMMU), configured to:
maintain a threshold compression ratio, wherein the threshold compression
ratio is a desired minimum ratio between compressed pages and uncompressed pages of the
system memory;
25 dynamically monitor a current compression ratio for the system memory,
wherein the current compression ratio is an actual ratio between compressed pages and
uncompressed pages of the system memory, and wherein the compression ratio determines
an amount by which system memory address space can be increased;
dynamically determine if the current compression ratio is below the
30 threshold compression ratio;

Sub
A3

if the current compression ratio is below the threshold compression ratio, cause the compression of one or more uncompressed pages to generate one or more compressed pages; and

wherein the CMMU is operable to increase the effective size of the system memory by keeping least recently used pages compressed in the physical memory and most recently and frequently used pages uncompressed in the physical memory; and

wherein compressing the one or more uncompressed page is operable to increase the current compression ratio for the system memory.

130. The system of claim 129, wherein the threshold compression ratio is programmable to determine an amount by which the effective size of the system memory can be increased

131. The system of claim 129, further comprising:

a compression engine;

wherein, in said causing the compression of the one or more uncompressed pages, the CMMU is further configured to write the uncompressed pages to the compression engine; and

wherein the compression engine is configured to compress the one or more uncompressed pages to produce the one or more compressed pages.

132. The system of claim 129, wherein, in said causing the compression of the one or more uncompressed pages, the CMMU is further configured to:

locate a page translation entry in a page translation table referencing an uncompressed page in the system memory;

determine that the uncompressed page is highly compressible;

set a highly compressed attribute in the page translation entry to indicate that the page is highly compressed; and

free a portion of the physical memory allocated to the page in the system memory;

Sub
A3

wherein highly compressing the uncompressed page is operable to increase the current compression ratio for the system memory.

133. The system of claim 132, wherein the highly compressed attribute is a one-bit field in the page translation entry.

134. The system of claim 129, wherein the CMMU is further configured to, prior to said freeing the portion of the physical memory allocated to the page:
determine if the page is dirty; and
if the page is dirty, write data from the page to non-volatile storage to make the page clean.